(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平6－266620

(43)公開日　平成６年(1994)９月22日

(51)Int.Cl.[5]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	12/08	３１０　Ａ	7608－5B		
	9/46	３４０　Ｆ	8120－5B		
	12/00	５７２	9366－5B		

審査請求　未請求　請求項の数２　ＯＬ　（全 15 頁）

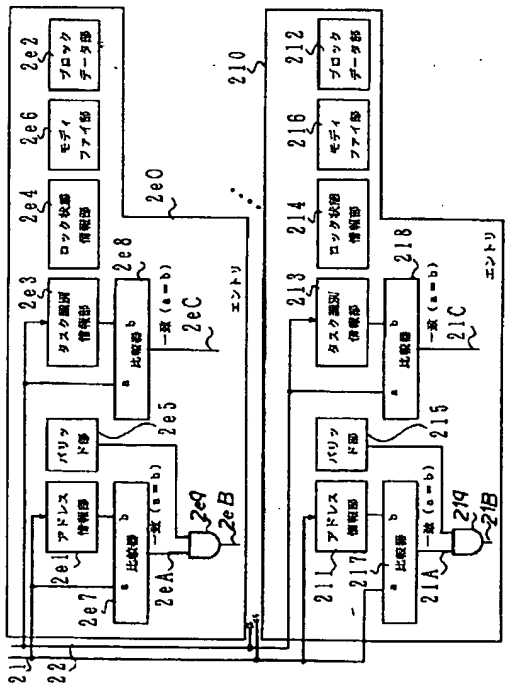
(21)出願番号　特願平5－55277

(22)出願日　平成５年(1993)３月16日

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目１番６号
(72)発明者　塩澤　恒道
東京都千代田区内幸町１丁目１番６号　日本電信電話株式会社内
(74)代理人　弁理士　磯村　雅俊

(54)【発明の名称】　キャッシュメモリを用いた排他制御システム

(57)【要約】
【目的】　マルチタスク処理における排他制御を効率良く行ない、共通データの無矛盾性の保証性能と、処理ユニットの実効的な性能とを向上させる。
【構成】　メインメモリのブロック単位のデータを各エントリに格納し、このエントリを介して、マルチタスク処理を行なう処理ユニットからのアクセスに対応するデータの転送制御と排他制御を行なうキャッシュメモリにおいて、エントリに格納されているブロックを排他制御の対象として設定した処理ユニットの各タスクの識別情報を登録するタスク識別情報登録部を、エントリごとに設ける構成とし、タスク単位で、エントリに格納されているブロックの排他制御、および、この排他制御の設定と解除を行なう。



1

【特許請求の範囲】

【請求項1】　タスクを分割して並列に処理する処理ユニットがアクセスするメインメモリに格納されているブロック単位のデータを、各エントリに格納し、該エントリを介して、上記処理ユニットからのアクセスに対するデータの転送制御を行なうと共に、上記処理ユニットで並列に処理される複数のタスクからの同一のエントリへのアクセスを制御してデータの矛盾を回避する排他制御を行なうキャッシュメモリにおいて、上記排他制御のために、上記エントリに、上記処理ユニットで並列に処理される各タスクの識別情報を登録するタスク識別情報登録手段と、上記エントリに格納されているブロックに対する排他制御情報を登録する排他制御情報登録手段を、上記エントリごとに設け、タスク単位で、上記エントリに格納されているブロックの排他制御、および、該排他制御の設定と解除を行なうことを特徴とするキャッシュメモリを用いた排他制御システム。

【請求項2】　請求項1に記載のキャッシュメモリを用いた排他制御システムにおいて、上記キャッシュメモリは、上記処理ユニットのタスクからの上記排他制御の解除指示に基づき、該タスクの識別情報を登録しているエントリに対する上記タスク識別情報の解除、および、該エントリに対する全てのタスク識別情報が解除された場合での上記排他制御情報の解除を行なうことを特徴とするキャッシュメモリを用いた排他制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数のタスクを並列に実行するマルチタスク処理を行なう処理ユニットからのアクセスに対するのデータの転送制御を行なうキャッシュメモリを用いた排他制御技術に係わり、特に、複数のタスク間でアクセスされるデータの無矛盾性の保証を、効率良く行なうのに好適なキャッシュメモリを用いた排他制御システムに関するものである。

【0002】

【従来の技術】一般に、メインメモリのデータ転送速度は、処理ユニットの高速な処理速度に比べて低速であり、このメインメモリの低速なデータ転送動作は、システムの性能低下の原因となっている。このような、問題を解決するためには、例えば、社団法人電子情報通信学会編「電子情報通信ハンドブック」（1988年、株式会社オーム社発行）の第1670頁から第1671頁に記載のように、キャッシュメモリ（バッファ記憶）を設けることが有効である。

【0003】このようなキャッシュメモリを介して、マルチタスク処理を行なう処理ユニットの各タスクが、共通にアクセスする共通データの読み出し、および、書き換えを行なう場合、他のタスクとの間に矛盾が発生する。例えば、タスクAとタスクBが、メインメモリのアドレスX（キャッシュメモリのエントリX）に格納している値「x」に、それぞれ、「10」および「100」を加算する場合、タスクAとタスクBの処理が矛盾なく行なわれると、アドレスXの値は「x＋110」となる。しかし、タスクAが、アドレスXの値「x」を読み出してからアドレスXに値「x＋10」を書き込む前に、処理ユニットで実行されるタスクが、タスクBに切り替わると、タスクBが、アドレスXの値「x」を読み出し、タスクBは、アドレスXを「x＋100」に書き換える。そして、タスクAが再開されると、タスクAは、アドレスXを「x＋10」に書き換え、最終的なアドレスXの値は「x＋10」となる。

2

【0004】このような矛盾が発生しないように、各タスクは、共通データにアクセスする前に、共通データへのアクセス権の獲得を行なう。このアクセス権の獲得が成功すると、タスクは、共通データへのアクセスを行ない、共通データへのアクセス終了後に、このアクセス権を解放する。従来技術では、このアクセス権は、共通データごとに設けられ、同時には1つのタスクにのみ、アクセス権が与えられるように制御される。

【0005】このように、従来のマルチタスクでのデータの矛盾を保証する制御においては、各タスクは、共通データにアクセスする前にアクセス権を獲得し、他のタスクからの共通データへのアクセスを抑止する制御や、あるタスクが共通データにアクセスしている間には、処理ユニットで実行中のタスクの切り替えを抑止する制御などの排他制御が必要である。一般に、このような排他制御では、対象となるデータを以下の3つの状態で管理する。

（a）リードロック状態：1つ以上のタスクによって読み出し中であり、他タスクからの書き換えアクセスを抑止する必要がある。

（b）ライトロック状態：1つのタスクによって書き換え中であり、他タスクからの読み出し、および、書き換えアクセスを抑止する必要がある。

（c）アンロック状態　：何れのタスクからも全てのアクセスが可能である。

【0006】しかし、このような排他制御技術では、各タスクが同時並列にアクセスする共通データごとに、アクセス権の確保と解放の処理を行なう必要があるため、このための処理量が増加する欠点があった。このような問題を解決するための従来技術として、同時並列にアクセスしないデータと共通データを厳密に分けずに、アクセス権の確保と解放を、複数の共通データに対して一括して行なうことにより、アクセス権の確保と解放の回数を減らす技術がある。

【0007】しかし、このように、複数の共通データに対して一括してアクセス権の確保と解放を行なうと、排他制御の対象となる共通データの対象が増加し、無用な排他制御が行なわれるため、複数のタスクで並列に行なう処理の実行が阻害される確率が大きくなってしまう。

また、タスクの切り替えを抑止する排他制御では、共通データへのアクセスを頻繁に行なうタスクの実行が優先されることとなり、例えば、タスクによっては、開始から終了まで切り替えが行なわれなくなり、タスクを相互に切り替えて分割処理することによって実行の並列化を図るマルチタスクの制御が複雑になってしまう。

【0008】

【発明が解決しようとする課題】解決しようとする問題点は、複数の共通データに対して一括してアクセス権の確保と解放を行なう従来の排他制御技術では、無用な排他制御が行なわれ、マルチタスク処理が阻害される確率が大きくなってしまい、また、タスクの切り替えを抑止する従来の排他制御技術では、共通データへのアクセスを頻繁に行なうタスクの実行が優先され、マルチタスクの並列処理制御が複雑になってしまい、排他制御を効率良く行なうことができない点である。本発明の目的は、これら従来技術の課題を解決し、キャッシュメモリのエントリへの各タスクのアクセスに対する排他制御を効率良く行ない、マルチタスク処理を行なう処理ユニットの実効的な性能を向上させることを可能とするキャッシュメモリを用いた排他制御システムを提供することである。

【0009】

【課題を解決するための手段】上記目的を達成するため、本発明のキャッシュメモリを用いた排他制御システムは、（1）タスクを分割して並列に処理する処理ユニットがアクセスするメインメモリに格納されているブロック単位のデータを、各エントリに格納し、このエントリを介して、処理ユニットからのアクセスに対するデータの転送制御を行なうと共に、処理ユニットで並列に処理される複数のタスクからの同一のエントリへのアクセスを制御してデータの矛盾を回避する排他制御を行なうキャッシュメモリにおいて、排他制御のために、エントリに、処理ユニットで並列に処理される各タスクの識別情報を登録するタスク識別情報登録部と、エントリに格納されているブロックに対する排他制御情報を登録する排他制御情報登録部を、エントリごとに設け、タスク単位で、エントリに格納されているブロックの排他制御、および、この排他制御の設定と解除を行なうことを特徴とする。また、（2）上記（1）に記載のキャッシュメモリを用いた排他制御システムにおいて、キャッシュメモリは、処理ユニットのタスクからの排他制御の解除指示に基づき、このタスクの識別情報を登録しているエントリに対するタスク識別情報の解除、および、このエントリに対する全てのタスク識別情報が解除された場合での排他制御情報の解除を行なうことを特徴とする。

【0010】

【作用】本発明においては、マルチタスク処理を行なう処理ユニットの各タスクの、キャッシュメモリの同一エントリに対する排他制御を、各タスク単位で行なう。例えば、キャッシュメモリの各エントリは、各エントリを識別するアドレス情報や、他のタスクからの全てまたは一部のアクセスを抑止する排他制御が設定された状態を示すロック状態情報などと共に、排他制御の設定（ロック）を指示したタスクの識別情報を登録する。そして、ロック状態としたエントリに対する抑止が必要なアクセスが発生した場合、このアクセスを行なったタスクを判別する。もし、他のタスクからのアクセスであれば、このタスク、もしくは、このエントリをロックしたタスクのいずれかを中断、または、無力化（中止）することにより、排他制御を行なう。このことにより、各タスクが並列にアクセスする共通なエントリのアクセス権の確保と解放の処理を行なうための処理量の増加を回避することができ、かつ、タスクの切り替えを抑止する制御も不要となる。

【0011】

【実施例】以下、本発明の実施例を、図面により詳細に説明する。図1は、本発明の排他制御システムに用いるキャッシュメモリの本発明に係わる構成の一実施例を示すブロック図であり、図2は、このキャッシュメモリを用いた情報処理システムの構成の一具体例を示すブロック図である。図2において、1は、分割した複数のタスクを並列に実行してマルチタスク処理を行なう処理ユニット、2は、本発明に係わる排他制御動作を行なうキャッシュメモリ、3は、処理ユニット1とキャッシュメモリ2間を接続するキャッシュバス、4は、処理ユニット1からのアクセスの対象となる元のデータを格納するメインメモリ、5は、キャッシュメモリ2とメインメモリ4間を接続するメモリバスである。

【0012】キャッシュバス3は、処理ユニット1から送出されるアドレスを伝達するためのアドレスバス31と、処理ユニット1とキャッシュメモリ2の間で授受されるデータを伝達するためのデータバス32と、処理ユニット1から送出されるアクセスの内容（読み出し／書き換え）を伝達するためのリード／ライトバス33と、本発明に係わり、処理ユニット1で実行中の各タスクの識別情報を伝達するためのタスク識別情報バス34と、処理ユニット1から送出されるロック（排他制御設定）指示情報を伝達するためのロック指示バス35と、キャッシュメモリ2から送出される処理ユニット1からのアクセスに対応する制御結果を伝達するための転送応答バス36と、キャッシュメモリ2から送出される処理ユニット1からのアクセスが矛盾した結果を伝達するための矛盾応答バス37とからなる。

【0013】処理ユニット1は、アドレスaで指定されるデータの読み出し、または、書き換えアクセスを行なうとき、アドレスバス31にアドレスaを出力し、リード／ライトバス33に、「読み出し」または「書き換え」の種別を出力し、タスク識別情報バス34に、処理ユニット1で実行中のタスクに予め割り当てられたタス

ク識別情報を出力し、ロック指示パス３５に、「ロック設定指示」または「ロック解除指示」を出力し、書き換えの場合には、データパス３２に、書き換えデータを出力する。

【００１４】本実施例においては、タスク識別情報パス３４は、複数の信号線からなり、各信号線には、処理ユニット１で並列して実行する各タスクが、重複しないように割り当てられている。そして、処理ユニット１で実行中のタスクは、キャッシュメモリ２にアクセスする際に、割り当てられている信号線のみに、値「１」を出力し、他の信号線には、値「０」を出力する。また、ロック情報の解除を指示するためのパスは設けず、処理ユニット１からのアクセス時に、ロック指示パス３５にロック指示が出力されていない場合に、当該するエントリに対するロック情報の解除を指示する。また、メインメモリ４は、同じ大きさのブロックに分割され、キャッシュメモリ２との間では、メモリパス５を介して、ブロック単位にメモリ内容の転送を行なう。本発明に係わるキャッシュメモリ２は、図１で示すように、メインメモリ４の各ブロック単位のメモリ内容を格納する複数のエントリからなる。

【００１５】図１において、キャッシュメモリ２のｅ個の各エントリ２１０～２ｅ０は、転送されたブロックのアドレス情報を格納するアドレス情報部２１１～２ｅ１と、ブロック単位のメモリ内容を格納するブロックデータ部２１２～２ｅ２と、図２のタスク識別情報パス３４の信号線と同一のビット数からなり、ロック指示を行なったタスクが値「１」を出力した信号線に対応するビットに、値「１」を設定する本発明のタスク識別情報登録部としてのタスク識別情報部２１３～２ｅ３と、ロック（排他制御）状態を示す本発明の排他制御情報登録部としてのロック状態情報部２１４～２ｅ４と、各エントリ２１０～２ｅ０に格納されている内容が有効であるか無効であるかを示すバリッド部２１５～２ｅ５と、ブロックデータ部２１２～２ｅ２の内容のみを書き換えたことを示すモディファイ部２１６～２ｅ６と、アドレス情報部２１１～２ｅ１と信号線２１との内容とを比較する比較器２１７～２ｅ７と、タスク識別情報部２１３～２ｅ３と信号線２２との内容とを比較する比較器２１８～２ｅ８と、比較器２１７～２ｅ７から「一致」した旨の信号が出力され、かつ、バリッド部２１５～２ｅ５の内容が、「１（有効）」である場合に、信号線２１Ｂ～２ｅＢに「オン」を出力するアンドゲート２１９～２ｅ９とからなる。

【００１６】以下、このような構成の本実施例のキャッシュメモリの本発明に係わる動作を、図２のメインメモリ４のアドレスａで指定されるデータを含むブロックをＢａとし、図２の処理ユニット１が、ブロックＢａをアクセスする場合を例として、図３～図８を用いて説明する。

【００１７】図３～図８は、図２における情報処理システムの本発明に係わる処理動作の第１の具体例を示すフローチャートである。本例は、主に、図２の処理ユニット１からのアクセスに対する図１における本実施例のキャッシュメモリ２の処理を示すものであり、論理回路を用いることにより、この処理の各部分を並列に実行することができる。まず、図３における処理動作の説明を行なう。図２の処理ユニット１が、図２のキャッシュパス３を介して図２のメインメモリ４のａ番地のブロックＢａにアクセスすると（ステップ３０１）、図１におけるキャッシュメモリ２は、図２のアドレスパス３１に出力されているアドレスａのアドレス情報を、図１の信号線２１に入力し、また、図２のタスク識別情報パス３４に出力されているタスク識別情報を、図１の信号線２２に入力する（ステップ３０２）。そして、それぞれ入力したアドレス情報とタスク識別情報とに対応して、以下のように処理を行なう。

【００１８】図２の処理ユニット１から、図２のドレスパス３１を介して要求されたアドレスａのデータを含むブロックＢａを格納する有効なエントリが、図１のキャッシュメモリ２に存在しない、すなわち、図１の信号線２１Ｂ～２ｅＢの出力が「オン」であるエントリが存在しない場合（ステップ３０３）、図１のキャッシュメモリ２は、図１のロック状態情報部２１４～２ｅ４が、「０（アンロック状態）」であるエントリから、予め定められた手続きによって、エントリを選択する（ステップ３０４）。以下、この選択したエントリをエントリＥと記述し、このエントリＥは、アドレス情報部２ｘ１、ブロックデータ部２ｘ２、タスク識別情報部２ｘ３、・・・で構成されているものとする。

【００１９】エントリＥのバリッド部２ｘ５が「１（有効）」、かつ、モディファイ部２ｘ６が「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」であれば（ステップ３０５）、図１のキャッシュメモリ２は、図２のメモリパス５を介して、エントリＥのアドレス情報部２ｘ１で指定される図２メインメモリ４のブロックを、ブロックデータ部２ｘ２で書き換える（ステップ３０６）。そして、図１のキャッシュメモリ２は、図２のメモリパス５を介して、図２のメインメモリ４からブロックＢａを読み出し、エントリＥのブロックデータ部２ｘ２に格納し、アドレスａのアドレス情報を、アドレス情報部２ｘ１に格納し、バリッド部２ｘ５を「１（有効）」として（ステップ３０７）、次の図４に示す処理を行なう。

【００２０】以下、図４における処理動作の説明を行なう。図２の処理ユニット１が、図２のロック指示パス３５に、ロック指示信号を出力していない場合には（ステップ３０８）、タスク識別情報部２ｘ３の全てのビットに「０」を設定し（ステップ３０９）、ロック状態情報部２ｘ４に、「０（アンロック状態）」を設定する（ス

テップ３１０）。また、ステップ３０８において、図２の処理ユニット１が、図２のロック指示パス３５に、ロック指示信号を出力している場合には、タスク識別情報部２ｘ３に、図２のタスク識別情報パス３５に出力されている値を設定する（ステップ３１１）。そして、図２のリード／ライトパス３３に、アクセス種別として、「書き換え」ではなく「読み出し」が出力されていれば（ステップ３１２）、ロック状態情報部２ｘ４に、「１（リードロック状態）」を設定し（ステップ３１３）、また、「書き換え」が出力されていれば、ロック状態情報部２ｘ４に、「２（ライトロック状態）」を設定して（ステップ３１４）、次の図５に示す処理を行なう。

【００２１】以下、図５における処理動作の説明を行なう。図２のリード／ライトパス３３に、アクセス種別として「書き換え」ではなく「読み出し」が出力されていれば（ステップ３１５）、エントリＥのモディファイ部を「０（ブロックデータ部のＢａとメインメモリのＢａが一致）」とすると共に、エントリＥのブロックデータ部に格納されているブロックＢａのアドレスａで指定されるデータを、図２のデータパス３２に出力する（ステップ３１６）。また、ステップ３１５において、図２のリード／ライトパス３３に、アクセス種別として「書き換え」が出力されていれば、エントリＥのモディファイ部２ｘ６を、「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」とすると共に、エントリＥのブロックデータ部に格納されているブロックＢａのアドレスａで指定されるデータを、図２のデータパス３２に出力されている内容で書き換える（ステップ３１７）。以上の処理が終了すると、図２のキャッシュメモリ２は、図２の転送応答パス３６を介して、図２の処理ユニット１に、図２のデータパス３２に読み出しデータを出力した旨、または、書き換えアクセスが完了した旨を通知する（ステップ３１８）。

【００２２】また、図３のステップ３０３において、図２の処理ユニット１から、図２のアドレスパス３１を介して要求されたアドレスａのデータを含むブロックＢａを格納する有効なエントリＥが、図２のキャッシュメモリ２に存在する場合、すなわち、図１における信号線２１Ｂ～２ｅＢの出力が「オン」であるエントリが存在する場合には、次の図６～図８で示す処理動作を行なう。

【００２３】まず、図６における処理動作の説明を行なう。本図６は、エントリＥのロック情報情報部２ｘ４が（ステップ３１９）、「０（アンロック状態）」の場合の処理を示している。図２の処理ユニット１が、図２のロック指示パス３５にロック指示信号を出力し（ステップ３２０）、かつ、図２のリード／ライトパス３３に、アクセス種別として、「書き換え」を出力しているとき（ステップ３２１）、エントリＥのモディファイ部２ｘ６が、「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」であれば、図２のキャッシュメモリ２は、図２のメモリパス５を介して、図２のメインメモリ４のブロックＢａを、ブロックデータ部２ｘ２の内容で書き換えると共に、モディファイ部２ｘ６を「０（ブロックデータ部のＢａとメインメモリのＢａが一致）」とする（ステップ３２２）。

【００２４】そして、ロック状態情報部２ｘ４に「２（書き換え／ライトロック状態）」を設定し（ステップ３２３）、タスク識別情報部２ｘ３に、図２のタスク識別情報パス３４に出力されている値を設定する（ステップ３２４）。また、ステップ３２１において、アクセス種別として、「書き換え」ではなく「読み出し」を出力しているときは、ロック状態情報部２ｘ４に「１（読み出しロック状態）」を設定し（ステップ３２５）、ステップ３２４の処理を行なう。このステップ３２４の処理のあとには、上述の図５に示す処理動作を行なう。

【００２５】次に、図７における処理動作の説明を行なう。本図７は、図６のステップ３１９において、エントリＥのロック情報情報部２ｘ４が「１（リードロック状態）」の場合の処理を示している。図２の処理ユニット１が、図２のロック指示パス３５に、ロック指示信号を出力し（ステップ３２６）、かつ、図２のリード／ライトパス３３にアクセス種別として「書き換え」を出力しているとき（ステップ３２７）、図２のタスク識別情報パス３４の全ての信号線の値と、エントリＥのタスク識別情報部２ｘ３の対応する挿げてのビット値とが一致しなければ、すなわち、図１の信号線２ｘｃの出力がオフであれば（ステップ３２８）、後述の図８におけるステップ３２９の処理を行なう。すなわち、図２のキャッシュメモリ２は、図２の矛盾応答パス３８を介して、矛盾したアクセス、すなわち、他のタスクがリードロック中のブロックに対して、書き換えアクセスの要求を行なった旨を通知し、当該アクセスを完了する。

【００２６】また、ステップ３２８において、図２のタスク識別情報パス３４の全ての信号線の値と、エントリＥのタスク識別情報部２ｘ３の対応する挿げてのビット値とが一致、すなわち、図１の信号線２ｘｃの出力がオンであり、かつ、エントリＥのモディファイ部２ｘ６が「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」であれば、図２のキャッシュメモリ２は、図２のメモリパス５を介して、図２のメインメモリ４のブロックＢａを、ブロックデータ部２ｘ２の内容で書き換え、モディファイ部２ｘ６を「０（ブロックデータ部のＢａとメインメモリのＢａが一致）」に設定する（ステップ３３０）。そして、ロック状態情報部２ｘ４に「２（書き換えロック状態）」を設定する（ステップ３３１）。この場合、図２のタスク識別情報パス３４の出力とエントリＥのタスク識別情報部２ｘ３の値とが等しく、ブロックＢｅをリードロック状態としたのは、処理ユニットで実行中のタスクのみである。

【００２７】また、ステップ３２７において、図２のリ

ード／ライトパス３３に、アクセス種別として「読み出し」を出力していれば、タスク識別情報部２ｘ３に、タスク識別情報部２ｘ３の値とタスク識別情報パス３４に出力されている値との論理和を設定する（ステップ３３２）。例えば、タスク識別情報部２ｘ３に、値「０１１０１０１０」が設定されており、タスク識別情報パス３４に、値「００００１００」が出力されている場合、タスク識別情報部２ｘ３には、値「０１１０１１１０」が設定される。

【００２８】また、ステップ３２６において、図２のロック指示パス３５にロック指示信号を出力していない場合、タスク識別情報部２ｘ３に、タスク識別情報部２ｘ３の値と、タスク識別情報パス３４に出力されている「０」と「１」を反転した値との論理積を設定する（ステップ３３３）。例えば、タスク識別情報部２ｘ３に値「０１１０１１１０」が設定されており、タスク識別情報パス３４に値「００００１００」が出力されている場合、タスク識別情報部２ｘ３には、「値「０１１０１０１０」が設定される。そして、タスク識別情報部２ｘ３の全てのビットが「０」（例えば、「００００００００」）となったときには（ステップ３３４）、ロック状態情報部２ｘ４に、値「０（アンロック状態）」を設定する（ステップ３３５）。このような処理のあとには、上述の図５に示す処理動作を行なう。

【００２９】次に、図８における処理動作の説明を行なう。本図８は、図６のステップ３１９において、エントリＥのロック情報情報部２ｘ４が「２（ライトロック状態）」の場合の処理を示している。図２の処理ユニット１が、図２のタスク識別情報パス３４に出力している内容と、エントリＥのタスク識別情報部２ｘ３の値とが等しくなければ、すなわち、図１における信号線２ｘｃの出力がオフであれば（ステップ３３６）、図２のキャッシュメモリ２は、図２の矛盾応答パス３８を介して、矛盾したアクセス、すなわち、他のタスクがリードロック中のブロックに対して、書き換えアクセスの要求を行なった旨を通知し（ステップ３２９）、当該アクセスを完了する。また、ステップ３３６において、図２のタスク識別情報パス３４の出力と、エントリＥのタスク識別情報部２ｘ３の値とが等しく、かつ、図２の処理ユニット１が、図２のロック指示パス３５にロック指示信号を出力していない場合（ステップ３３７）には、タスク識別情報部２ｘ３の全てのビットに「０」を設定し、ロック状態情報部２ｘ４に「０（アンロック状態）」を設定する（ステップ３３８）。そして、以下、上述の図５に示す処理動作を行なう。

【００３０】尚、ステップ３２９における矛盾したアクセスを行なった旨の通知を受けると、図２の処理ユニット１は、実行中のタスクを中断して、予め定められた時間が経過したあとに、タスクを再開するか、もしくは、実行中のタスクが書き換えたデータの内容を、タスクが実行を開始する時点の状態に戻す処理を行なって、実行中の処理を無効化（中止）し、予め定められた時間が経過したあとに処理を最初から実行する。このような、実行中のタスクの処理の無効化（中止）は、無効化する図１におけるタスク識別情報部２１３～２ｅ３の値が等しく、かつ、図１のモディファイ部２１６～２ｅ６に、値「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」が設定されている全てのエントリのバリッド部２１５～２ｅ５に、値「０（無効）」を設定することによって行なうことができる。

【００３１】無効化するタスク識別情報と、図１のタスク識別情報部２１３～２ｅ３の値が等しく、かつ、図１のモディファイ部２１６～２ｅ６に、値「１（ブロックデータ部のＢａとメインメモリのＢａが不一致）」が設定されているエントリに格納されているブロックの内容は、無効化するタスクによって書き換えられる前に、図２のメインメモリ４に書き込まれているので、上述の処理を終了したあと、他のタスクは、書き換えられる前のブロックにアクセス可能となる。

【００３２】また、必ずしも、図２の矛盾応答パス３７を介して矛盾が発生した旨の通知を受信したときに実行中のタスクを中断、または、無効化せず、図２の処理ユニット１が、原因となった当該ブロックを格納しているエントリのタスク識別情報部の内容を読みだす処理部を設け、当該ブロックをロックしているタスクを無効化するものでも良い。これにより、共通データに並列にアクセスするタスクの中で、重要度が高く、他のタスクに比べて優先して実行させたい優先度の高いタスクと、優先度の低いタスクとの間で、共通データに対して競合が発生した場合、予め、タスクごとに割り当てられた優先度によって、優先度の低い処理を中止させ、優先度の高い処理を優先して実行させることが可能となる。

【００３３】以上のように、図１に示す実施例のキャッシュメモリでは、処理ユニットで実行するタスクを相互に切り替えて実行するマルチタスク処理において、複数のタスクが並列にアクセスする共通データのアクセス権の確保と開放（ロックとアンロック）の処理を、各タスクが共通データにアクセスする処理と同時に行なうことができる。このことにより、共通データのアクセス権の確保と開放のためのタスクの処理量の増加が生じず、また、タスクの切り替えを抑止することもなく、排他制御のための実効的な処理性能の低下をなくすことが可能となる。

【００３４】上述の実施例において、図２の処理ユニット１で実行中のタスクからの共通データへのアクセスが不要となったとき、図２の処理ユニット１は、図２のキャッシュメモリ２に対して、当該タスクのタスク識別情報とロック状態の解除を指示する。このような指示が入力されると、図１のキャッシュメモリ２は、次の図９で示すロック解除の処理を行なう。

【００３５】図９は、図２における情報処理システムの本発明に係わる処理動作の第２の具体例を示すフローチャートである。本例は、図２の処理ユニット１からのロック解除指示に対する図１のキャッシュメモリ２のロック解除の処理動作を示すものであり、このキャッシュメモリ２内には、エントリを順次指定可能なカウンタを設けている。そして、論理回路を用いることにより、この処理の各部分を並列に実行することができる。図１のキャッシュメモリ２は、図２のキャッシュバス３を介して、ロック解除の指示（予め定められたアドレスへのアクセス）と共に、タスク識別情報バス３４を介して、タスク識別情報（このタスク識別情報と共に出力されたロック指示によって設定された全てのエントリがロック解除処理の対象となる）を受信すると、エントリを順次指定可能なカウンタに初期値（ここでは「１」）を設定する（ステップ４０１）。このカウンタ値が図１のキャッシュメモリ２内のエントリ総数ｅになるまで、カウンタの値が＋１されるごとに、以下の処理を行なう。尚、ここで、カウンタの値「ｃ」で指定されるエントリをエントリＥｃとし、このエントリＥｃは、アドレス情報部２ｃ１、ブロックデータ部２ｃ２、タスク識別情報部２ｃ３、・・・で構成されているものとする。

【００３６】エントリＥｃのバリッド部２ｃ５の値が「０（無効）」（ステップ４０２）、または、ロック状態情報部２ｃ４の値が「０（アンロック状態）」の場合（ステップ４０３、４０４）、カウンタの値を＋１する（ステップ４１０）。ステップ４０２において、エントリＥｃのバリッド部２ｃ５の値が「１（有効）」で、かつ、ステップ４０３、４０４において、ロック状態情報部２ｃ４の値が「１（リードロック状態）」の場合には、タスク識別情報部２ｘ３に、タスク識別情報部２ｘ３の値と、タスク識別情報バス３４に出力されている「０」と「１」を反転した値との論理積を設定する（ステップ４０５）。例えば、タスク識別情報部２ｘ３に値「０１１０１１１０」が設定されており、タスク識別情報３４に値「０００００１００」が出力されている場合、タスク識別情報部２ｘ３には、値「０１１０１０１０」が設定される。タスク識別情報部２ｘ３の全てのビットが「０」（例えば、「００００００００」）となったときには（ステップ４０６）、ロック状態情報部２ｘ４に値「０（アンロック状態）」を設定し（ステップ４０７）、カウンタの値を＋１する（ステップ４１０）。

【００３７】また、ステップ４０２において、エントリＥｃのバリッド部２ｃ５の値が「１（有効）」で、ステップ４０３において、ロック状態情報部２ｃ４の値が「２（ライトロック状態）」の場合で、かつ、エントリＥｃのタスク識別情報部２ｃ３とタスク識別情報バス３４の値が等しい場合、すなわち、信号線２ｃＣの値がオンの場合（ステップ４０８）、バリッド部２ｃ５に値「０（無効）」をセットし（ステップ４０９）、リード／ライトバス３３の値を＋１する（ステップ４１０。ステップ４０８において、エントリＥｃのタスク識別情報部２ｃ３とタスク識別情報バス３４の値が等しくない場合、すなわち、信号線２ｃＣの値がオフの場合、そのまま、カウンタの値を＋１する（ステップ４１０）。

【００３８】このようにして、図２の処理ユニット１で実行するタスクにおいて、リードロック状態、または、ライトロック状態となっている共通データへのアクセスが不要となった場合、タスクが設定した排他制御情報を解除するためのアクセス（すなわち、ロック指示を行なわずに、当該ブロックへのアクセスを行なう）を、１回で行なうことができ、従来技術のように、共通データを格納しているブロックごとに行なうことが不要となり、ロック解除のための処理ユニットからのアクセス回数を削減することができる。特に、矛盾発生の通知を受信した場合の無効化のために、全てのロック状態を解除する場合に有効である。また、ロック状態の解除の指示と共に通知するタスク識別情報で、全てのタスクを指定する（例えば、「１１１１１１１１」を出力）ことによって、図１におけるキャッシュメモリ２の全てのロック情報を解除することも可能となる。

【００３９】以上、図１～図９を用いて説明したように、本実施例のキャッシュメモリでは、タスクが並列にアクセスする共通データへのアクセスを行なう際に、アクセス権の確保と開放の処理を、各処理ユニットで実行するアクセスと同時に実行することができる。このことにより、タスク間でアクセスするデータの無矛盾性を、効率良く保証することが可能である。また、タスクの並列処理を阻害する無用な排他制御、および、タスクの切り替えを抑止する制御などが不要となり、アクセス権の確保と開放のための処理ユニットのオーバヘッドも生じず、処理ユニットでの実効的な性能を向上させることが可能となる。尚、本発明は、図１～図９を用いて説明した実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。

【００４０】

【発明の効果】本発明によれば、キャッシュメモリのエントリに格納されているブロックへの各タスクのアクセスに対する排他制御を効率良く行なうことができ、タスク間でアクセスするデータの無矛盾性の保証処理の高効率化と、マルチタスク処理を行なう処理ユニットの実効的な性能の向上が可能である。

【図面の簡単な説明】

【図１】本発明の排他制御システムに用いるキャッシュメモリの本発明に係わる構成の一実施例を示すブロック図である。

【図２】図１におけるキャッシュメモリを用いた情報処理システムの構成の一具体例を示すブロック図である。

【図３】図２における情報処理システムの本発明に係わる処理動作の第１の具体例を示すフローチャートの第１

の部分である。

【図4】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第2の部分である。

【図5】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第3の部分である。

【図6】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第4の部分である。

【図7】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第5の部分である。

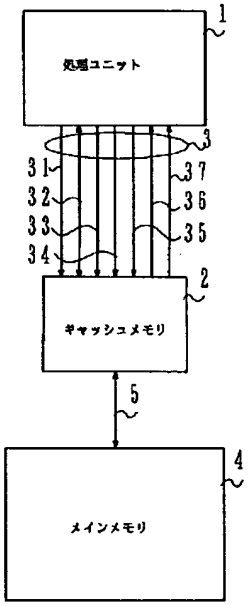
【図8】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第6の部分である。

【図9】図2における情報処理システムの本発明に係わる処理動作の第2の具体例を示すフローチャートである。

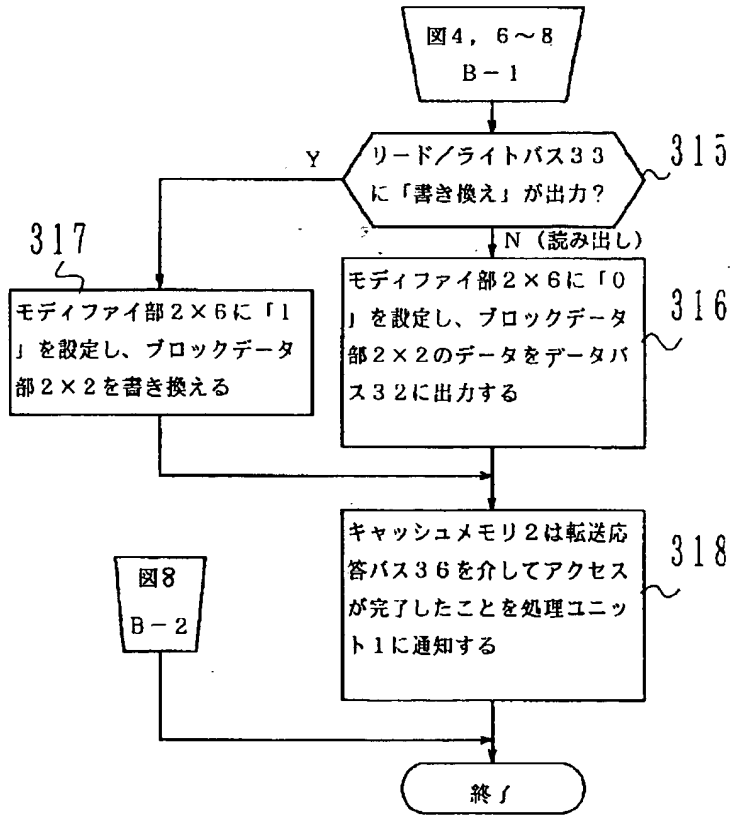
【符号の説明】

1　処理ユニット
2　キャッシュメモリ
3　キャッシュバス
4　メインメモリ
5　メモリバス
21、22　信号線
31　アドレスバス
32　データバス
33　リード／ライトバス
34　タスク識別情報バス
35　ロック指示バス
36　転送応答バス
37　矛盾応答バス
210～2e0　エントリ
211～2e1　アドレス情報部
212～2e2　ブロックデータ部
213～2e3　タスク識別情報部
214～2e4　ロック状態情報部
215～2e5　バリッド部
216～2e6　モディファイ部
217～2e7　比較器
218～2e8　比較器
219～2e9　アンドゲート
21B～2eB　信号線

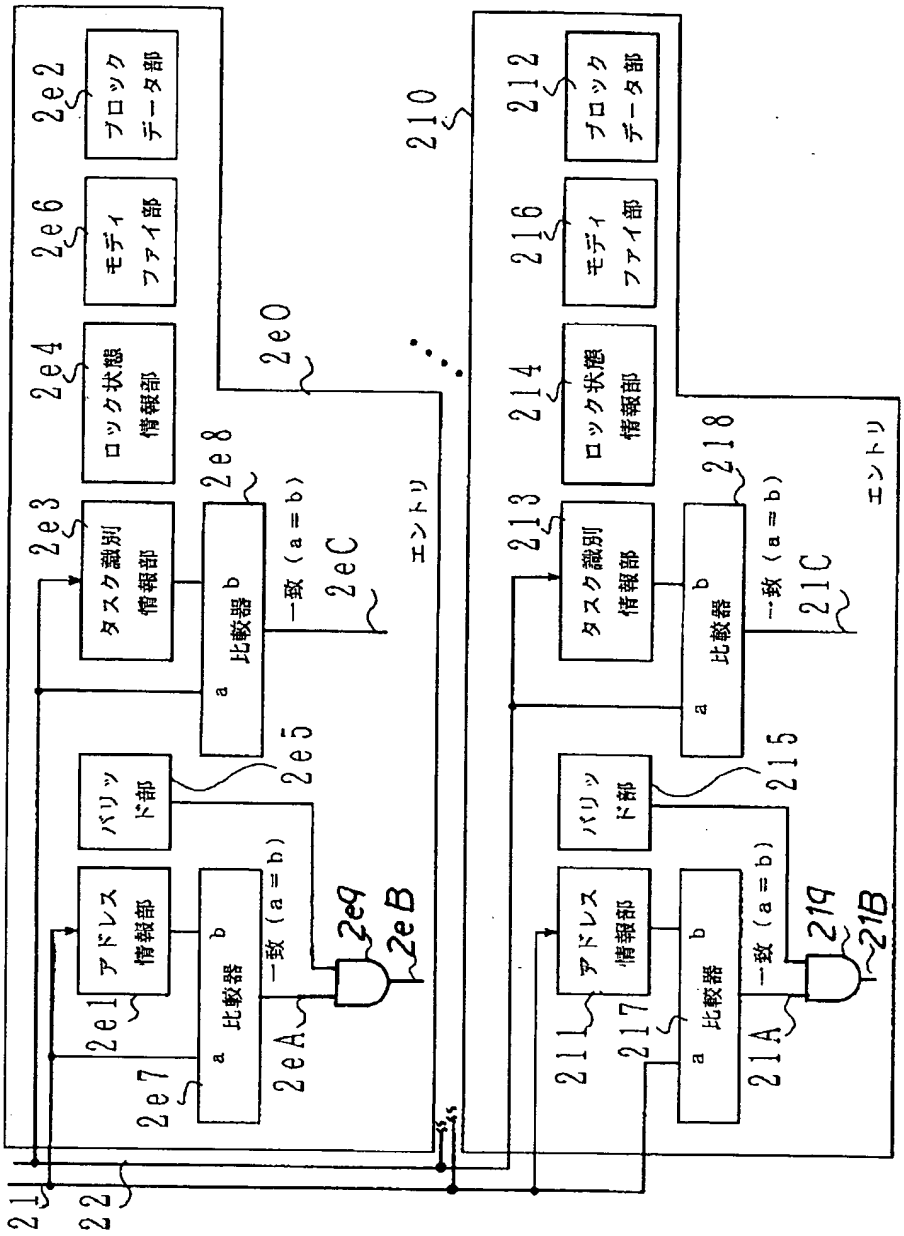
【図2】



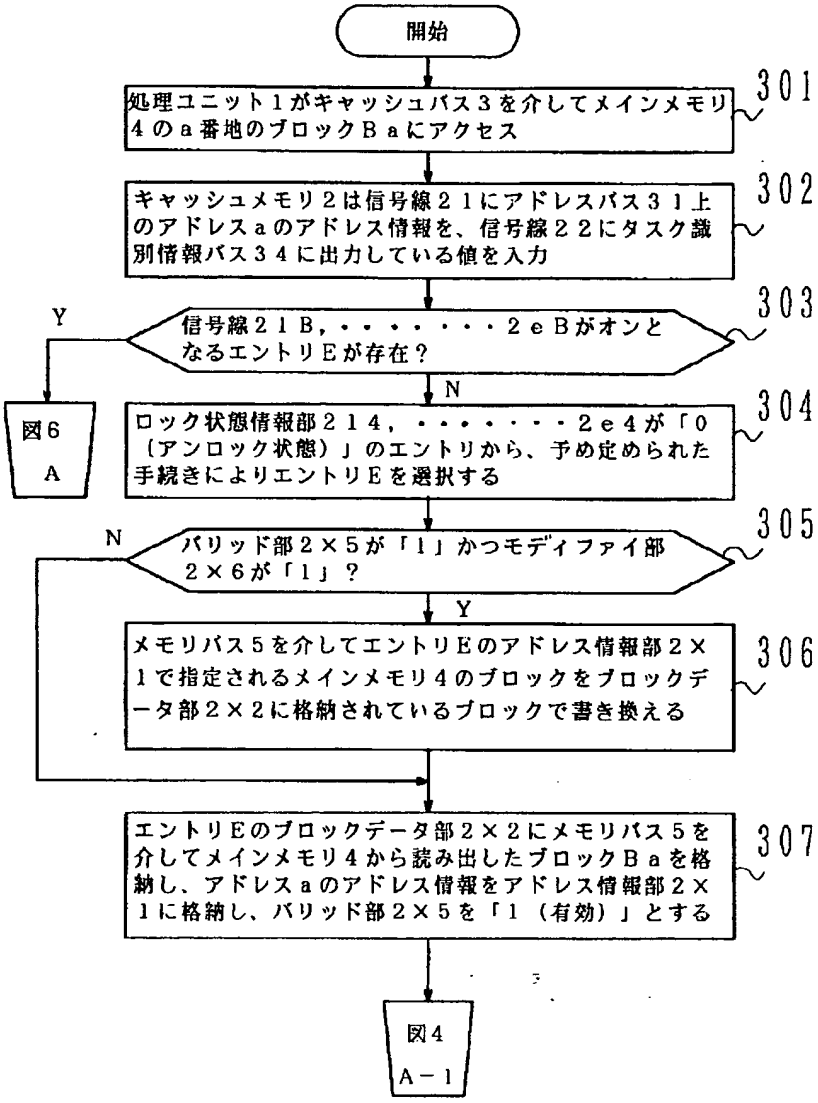
【図5】



【図1】


ブロック データ部
モディ ファイ部
ロック状態 情報部
タスク識別 情報部
比較器
一致（a＝b）
パリッド部
アドレス 情報部
エントリ
2e2
2e6
2e4
2e3
2e8
2e0
2eC
2e5
2e9
2eB
2e1
2e7
2eA
210
212
216
214
213
218
215
219
21B
211
217
21A
21C
21
22

【図3】


開始
処理ユニット１がキャッシュバス３を介してメインメモリ４のａ番地のブロックＢａにアクセス
301
キャッシュメモリ２は信号線２１にアドレスバス３１上のアドレスａのアドレス情報を、信号線２２にタスク識別情報バス３４に出力している値を入力
302
信号線２１Ｂ，・・・・・・・・２ｅＢがオンとなるエントリＥが存在？
303
Y
N
図６ Ａ
ロック状態情報部２１４，・・・・・・・・２ｅ４が「０（アンロック状態）」のエントリから、予め定められた手続きによりエントリＥを選択する
304
N
バリッド部２×５が「１」かつモディファイ部２×６が「１」？
305
Y
メモリバス５を介してエントリＥのアドレス情報部２×１で指定されるメインメモリ４のブロックをブロックデータ部２×２に格納されているブロックで書き換える
306
エントリＥのブロックデータ部２×２にメモリバス５を介してメインメモリ４から読み出したブロックＢａを格納し、アドレスａのアドレス情報をアドレス情報部２×１に格納し、バリッド部２×５を「１（有効）」とする
307
図４ Ａ－１

【図4】

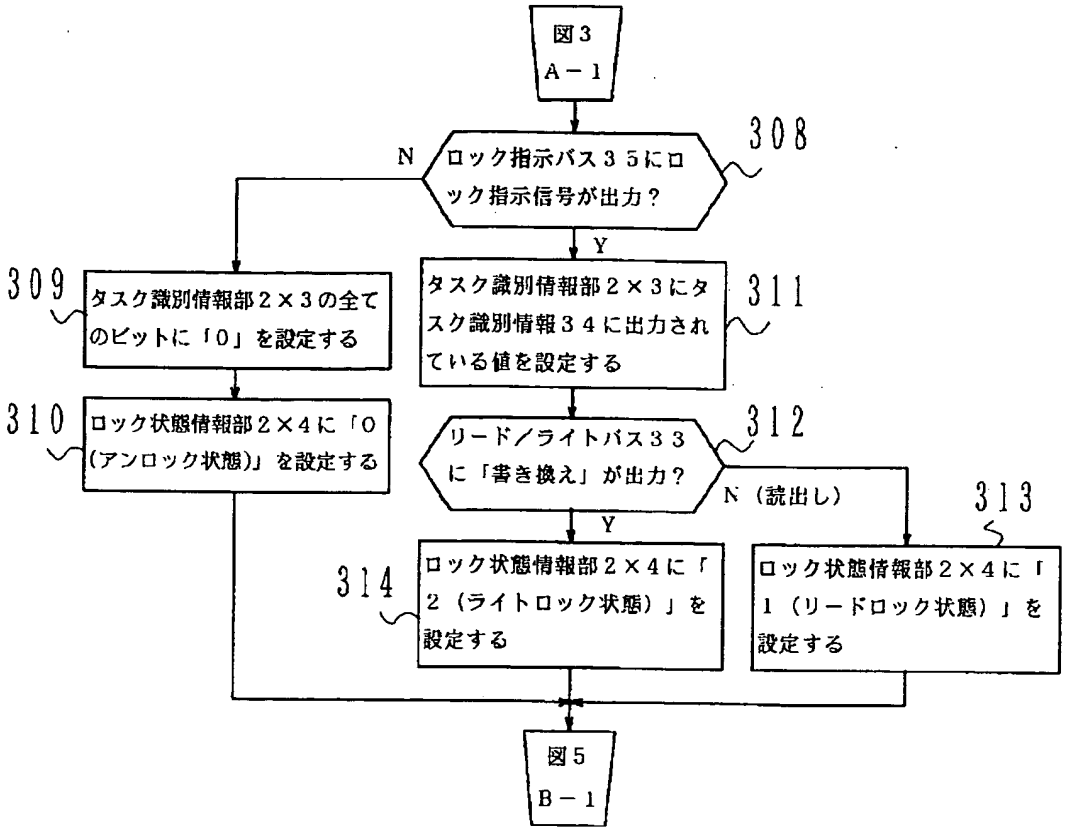


図3
A－1
308
N
ロック指示バス35にロック指示信号が出力？
Y
309
タスク識別情報部2×3の全てのビットに「0」を設定する
311
タスク識別情報部2×3にタスク識別情報34に出力されている値を設定する
310
ロック状態情報部2×4に「0（アンロック状態）」を設定する
312
リード／ライトバス33に「書き換え」が出力？
N（読出し）
313
Y
314
ロック状態情報部2×4に「2（ライトロック状態）」を設定する
ロック状態情報部2×4に「1（リードロック状態）」を設定する
図5
B－1

【図6】

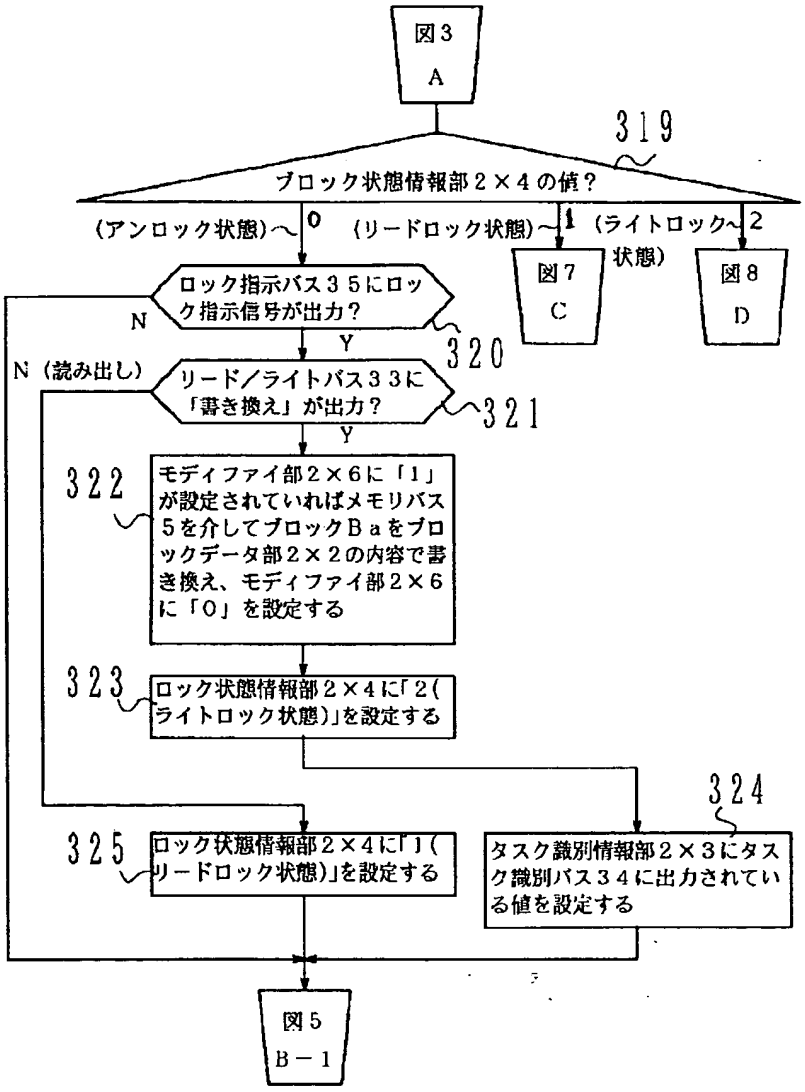


図3
A
319
ブロック状態情報部２×４の値？
(アンロック状態)
0
(リードロック状態)
1
(ライトロック状態)
2
図7
C
図8
D
ロック指示バス３５にロック指示信号が出力？
N
Y
320
N (読み出し)
リード／ライトバス３３に「書き換え」が出力？
321
Y
322
モディファイ部２×６に「１」が設定されていればメモリバス５を介してブロックＢａをブロックデータ部２×２の内容で書き換え、モディファイ部２×６に「０」を設定する
323
ロック状態情報部２×４に「２(ライトロック状態)」を設定する
324
タスク識別情報部２×３にタスク識別バス３４に出力されている値を設定する
325
ロック状態情報部２×４に「１(リードロック状態)」を設定する
図5
B－1

【図7】

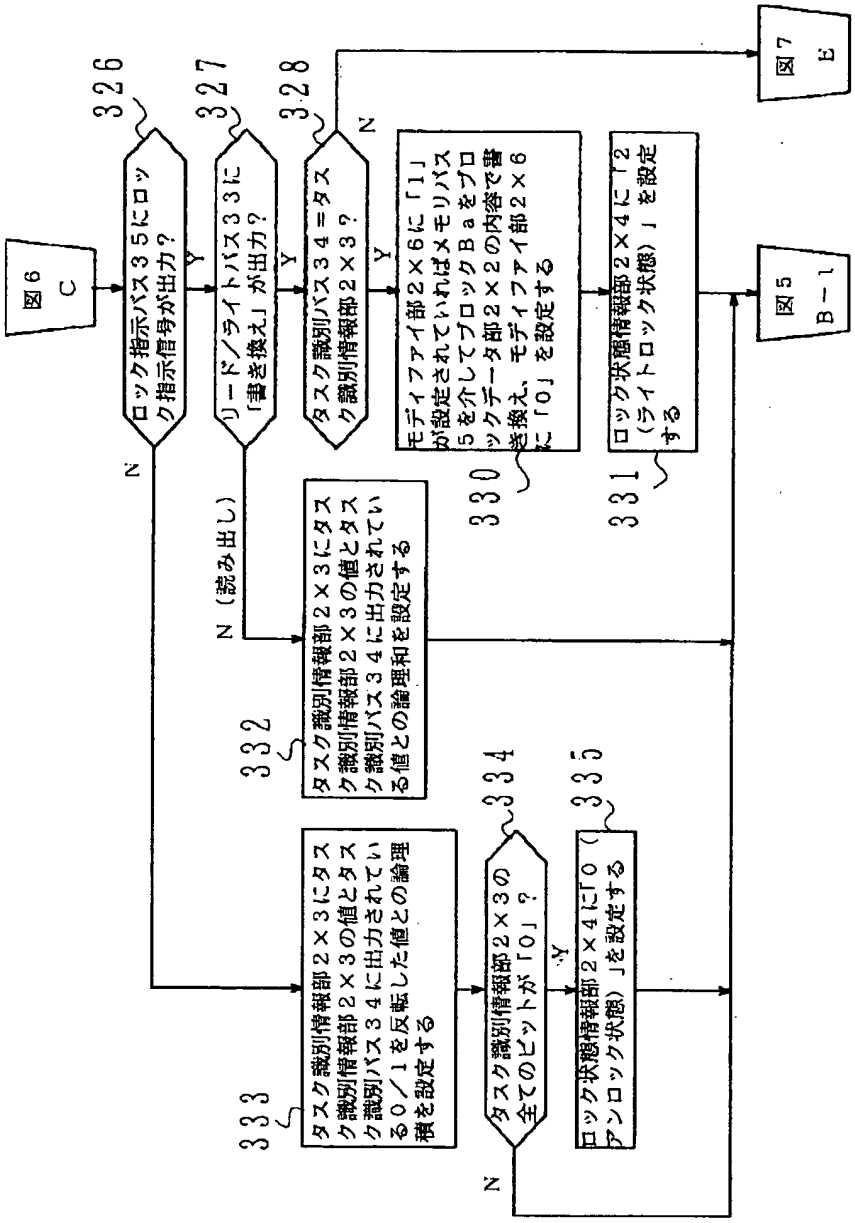


図6 C
326
ロック指示バス35にロック指示信号が出力？
Y
327
リードノライトバス33に「書き換え」が出力？
Y
328
タスク識別バス34＝タスク識別情報部2×3？
Y
N
図7 E
モディファイ部2×6に「1」が設定されていればメモリバス5を介してブロックBaをブロックデータ部2×2の内容で書き換え、モディファイ部2×6に「0」を設定する
330
ロック状態情報部2×4に「2（ライトロック状態）」を設定する
331
図5 B－1
N
N（読み出し）
タスク識別情報部2×3にタスク識別情報部2×3の値とタスク識別バス34に出力されている値との論理和を設定する
332
タスク識別情報部2×3にタスク識別情報部2×3の値とタスク識別バス34に出力されている0／1を反転した値との論理積を設定する
333
334
タスク識別情報部2×3の全てのビットが「0」？
Y
N
335
ロック状態情報部2×4に「0（アンロック状態）」を設定する

【図8】

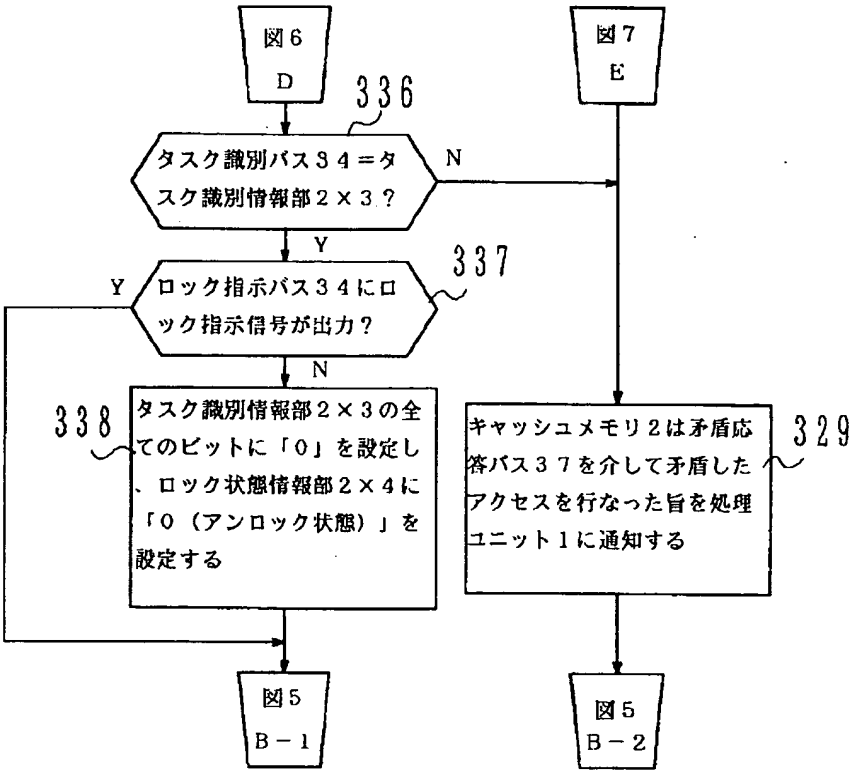
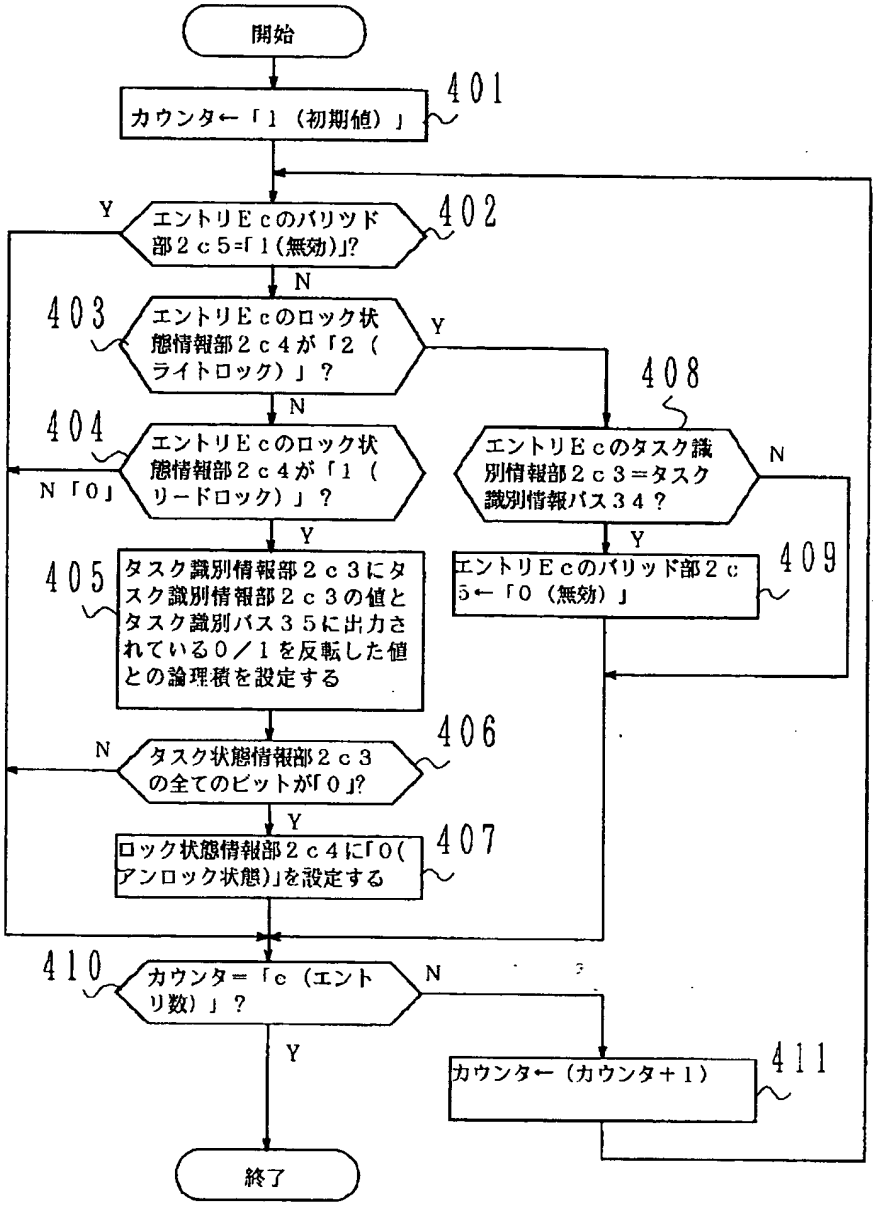


図6
D
336
タスク識別バス３４＝タスク識別情報部２×３？
N
Y
337
Y
ロック指示バス３４にロック指示信号が出力？
N
338
タスク識別情報部２×３の全てのビットに「0」を設定し、ロック状態情報部２×４に「0（アンロック状態）」を設定する
図5
B－1
図7
E
キャッシュメモリ２は矛盾応答バス３７を介して矛盾したアクセスを行なった旨を処理ユニット１に通知する
329
図5
B－2

【図9】


開始
401
カウンタ←「1（初期値）」
402
エントリEcのバリッド部2c5=「1(無効)」?
Y
N
403
エントリEcのロック状態情報部2c4が「2（ライトロック）」？
Y
N
404
エントリEcのロック状態情報部2c4が「1（リードロック）」？
N「0」
Y
405
タスク識別情報部2c3にタスク識別情報部2c3の値とタスク識別バス35に出力されている0／1を反転した値との論理積を設定する
406
タスク状態情報部2c3の全てのビットが「0」?
N
Y
407
ロック状態情報部2c4に「0(アンロック状態)」を設定する
408
エントリEcのタスク識別情報部2c3＝タスク識別情報バス34？
N
Y
409
エントリEcのバリッド部2c5←「0（無効）」
410
カウンタ＝「c（エントリ数）」？
N
Y
411
カウンター（カウンタ＋1）
終了